

## บทที่ 2

# ทฤษฎีที่เกี่ยวข้องและเทคโนโลยีที่ใช้

### 2.1 ยูนิตี้ เกม เอนจิน (Unity Game Engine)

ยูนิตี้ (Unity) คือเครื่องมือสร้างเกมแบบ 2 มิติ และ 3 มิติ สามารถทำงานบนระบบปฏิบัติการได้ 2 ระบบคือ Windows และ OSX ผลิตเกมออกไปได้หลากหลายแพลตฟอร์ม เช่น แอนดรอยด์, iOS, PC เป็นต้น Unity มีจุดเด่นคือมองทุกอย่างเป็นวัตถุ และในวัตถุต่างๆ ก็จะมีองค์ประกอบของวัตถุนั้นที่ทำงานร่วมกัน ไม่ว่าจะเป็นสิ่งของ ไอเทมในเกม ตัวละคร ทุกอย่างต้องการส่วนประกอบที่ทำให้ วัตถุเหล่านั้นเคลื่อนไหว หรือมีเสียงประกอบ นอกจากนั้น Unity มีร้านค้าออนไลน์สำหรับขาย Asset สำเร็จรูป ภาพพื้นหลังในเกม เสียงประกอบต่างๆ ชื่อว่า Asset Store รวมถึงคู่มือในการใช้งานที่ มีความละเอียด ง่ายต่อการศึกษาการใช้งาน นอกจากนั้น Unity มีชุมชนผู้พัฒนาจากทั่วโลก ที่พร้อมให้คำปรึกษาปัญหาที่เกิดขึ้นภายในการพัฒนา หรือศึกษาจากกรณีตัวอย่างจากกรณีที่เคยเกิดขึ้นได้โดยง่าย ถ้าต้องการใช้ในเชิงพาณิชย์จำเป็นต้องมีค่าใช้จ่าย ค่าลิขสิทธิ์ของโปรแกรมเพิ่มเติมด้วย



รูปที่ 2.1 โลโก้โปรแกรมยูนิตี้

ส่วนประกอบหลักของโปรแกรมนี้นี้

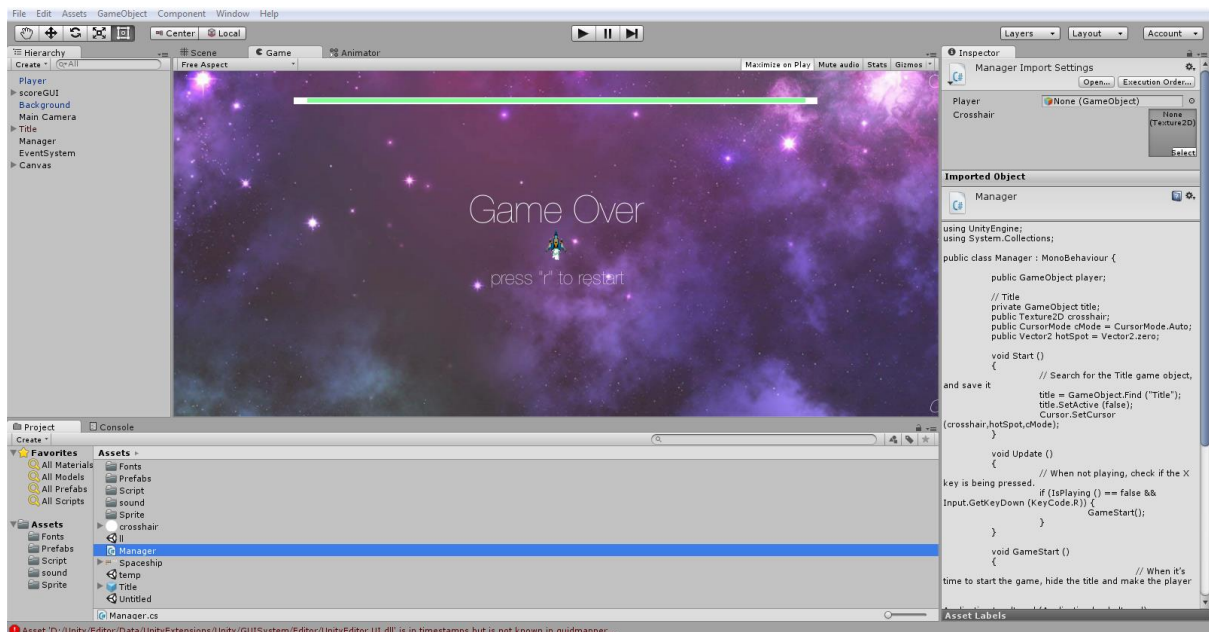
**แท็บเครื่องมือ** ประกอบด้วยเครื่องมือสำหรับเปลี่ยนมุมมองกล้อง ย้าย หมุน ย่อหรือขยายวัตถุ และปุ่มควบคุมการรันเกมในโหมด Debug

**แท็บมุมมอง** แบ่งเป็นสองแท็บย่อยคือ แท็บซีน (Scene) สำหรับมุมมองที่เห็นขณะออกแบบเกม และแท็บเกม (Game) สำหรับมุมมองที่ผู้เล่นเห็นจริงขณะเล่นเกม

**แท็บอินสเปกเตอร์ (Inspector)** ใช้แสดงค่าคุณสมบัติต่างๆ ของวัตถุที่กำลังทำงานอยู่

**แท็บลำดับชั้น (Hierarchy)** แสดงรายการวัตถุที่กำลังปรากฏอยู่ในฉากที่กำลังแสดงอยู่

**แท็บโปรเจกต์ (Project)** แสดงรายการวัตถุ โค้ด โปรแกรม ฉาก ไฟล์อ้างอิง และอื่นๆ ที่อยู่ในโปรเจกต์ โดยแท็บแต่ละแท็บนั้นสามารถปรับขนาดหรือลากย้ายไปยังตำแหน่งอื่นๆ ในหน้าจอของโปรแกรมยูนิตี้ได้



รูปที่ 2.2 หน้าตาของโปรแกรมยูนิตี้

## 2.2 The Art of Screen Shake โดย Jan Willem Nijman จากสตูดิโอ Vlambeer

Jan Willem Nijman จากสตูดิโอ Vlambeer ผู้พัฒนาเกมเช่น Nuclear Throne, Ridiculous Fishing, Luftrausers, etc ได้นำเสนอแนวคิดเกี่ยวกับ User Feedback ของ Action Game โดยโอเดียหลัก คือ การติดต่อกันระหว่างเกมและผู้เล่น โดยมีประเด็นหลักๆคือ ผู้เล่นนั้นควรจะสามารถที่จะ input ติดต่อกับ เกมได้อย่างต่อเนื่อง และ รวดเร็ว ตัวเกมนั้นควรสามารถ ตอบรับ ได้ตอบ และ แสดงผล input ของผู้เล่นได้ อย่างทันที โดยในประเด็นของตัวเกมนั้นจะแตกย่อยออกไปในเรื่องของ จำนวนศัตรู กล้องตามตัวผู้เล่น Sfx และ feedback ของการกระทำต่างๆ

โดยตัวอย่าง ที่ถูกยกมานั้นเป็นเกมแนว 2D Action Platformer ซึ่งมีข้อแนะนำดังต่อไปนี้

### 01 Basic Animation & Sfx

**02 Lower Enemy Hp:** HP ของศัตรูนั้นไม่จำเป็นต้องมากนักเนื่องจาก การสร้างศัตรูที่ HP มากๆขึ้นมาทำให้ ผู้เล่นต้องเสียเวลาอยู่กับศัตรูตัวเดียว จะทำให้เกมน่าเบื่อ ต่างกับการไล่ศัตรู HP น้อยเข้าไปเยอะๆทำให้มี เหตุการณ์เกิดขึ้นตลอดเวลา ทำให้ผู้เล่นต้อง input ต่างๆกันไปตลอดเวลา

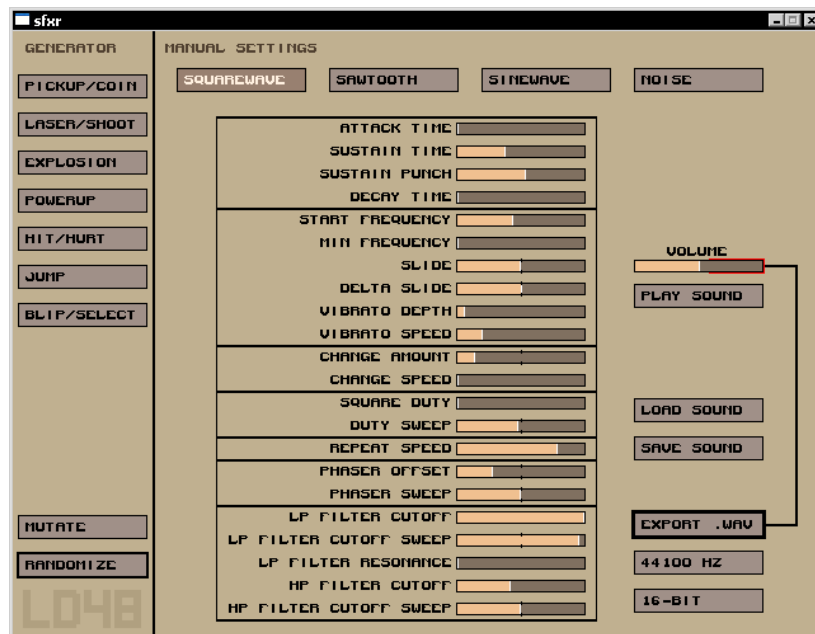
**03 Higher Rate of Fire:** ปืนที่ดีไซน์ควรจะมีรัวๆได้ เพราะวิธีการติดต่อกับโลกของเกมของผู้เล่นนั้น มี เพียงการยิง เพราะฉะนั้นเมื่อสามารถ input ได้รัวๆก็จะทำให้ user สามารถ input ได้ต่อเนื่องไม่ติดขัด

**04 More Enemies:** เมื่อสามารถยิงต่อเนื่องได้ จำนวนศัตรูควรเยอะขึ้นเพื่อให้เกมมีความสมดุล ไม่ง่ายเกินไป

- 05 Muzzle Flash:** มีแสงปลายกระบอกปืน เพื่อสื่อต่อผู้เล่นว่าได้เกิด Action ขึ้นต่อจาก Input ที่ได้รับมา
- 06 Faster Bullet:** เมื่อกระสุนเร็ว ทำให้ผู้เล่นสามารถที่จะ input ได้เร็วขึ้น
- 07 Less Accuracy:** ควรทำให้ปืนยิงออกมาไม่แม่นยำเพื่อเพิ่มความท้าทายให้กับผู้เล่น และ ดึงความสนใจของผู้เล่นไว้
- 08 Impact Effect:** เพื่อให้ผู้เล่นสามารถเห็นได้ชัดเจนว่า กระสุนที่ยิงไปนั้นตกกระทบกับอะไร
- 09 Hit Animation:** ให้รู้ว่าเกิดอะไรขึ้นเมื่อ Input ของ User ได้ส่งผลต่อโลกในเกมนั้นๆ
- 10 Enemy Knockback:** เพราะเมื่อศัตรูถูกยิง ควรมีสถานะบอกให้ผู้เล่นรับทราบว่าที่ยิงไปนั้นกระทบกับตัวศัตรู
- 11 Permanence:** เมื่อฆ่าศัตรูแล้ว ควรเหลือซากทิ้งไว้เพื่อแสดงให้เห็นถึงผลของการกระทำของผู้เล่นที่ได้ทำไป
- 12 Camera Lerp:** กล้องหน่วงค่อยๆเลื่อนตามผู้เล่น
- 13 Camera Position:** กล้องไม่ได้โฟกัสที่ตัวผู้เล่นแต่ โฟกัสที่ Action ที่ผู้เล่นกระทำตามจุดประสงค์ของเกม
- 14 Screenshake:** Screenshake เป็นการโต้ตอบกลับไปสู่ผู้เล่นที่สำคัญอย่างหนึ่งโดยสามารถใช้เป็นตัวบ่งบอกว่าตัวผู้เล่นกำลังตกอยู่ในอันตรายได้
- 15 Player Knockback:** เราควรให้เหตุผลกับผู้เล่นในการที่จะไม่กดปุ่มโจมตีค้าง มิฉะนั้นควรดีไซน์ให้ไม่ต้องกดปุ่มในการยิงไปเลย
- 16 Sleep:** สั่งให้เกมหยุดตัวเอง ไม่กี่ ms โดยตัวผู้เล่นนั้นจะไม่สังเกตเห็นชัดเจน แต่สมอนั้นจะได้มีเวลาในการตอบสนองต่อสิ่งที่ AI ศัตรูจะทำมากขึ้น
- 17 More Bass in SFX:** เพราะเสียงเป็นหนึ่งในส่วนสำคัญในการตอบสนองต่อ input ของผู้เล่น โดยมี case study เรื่องเสียงปืนของเกม Wolfenstein (2000)
- 18 Super Machine Gun:** เพิ่มความหลากหลายใน Input ที่ผู้เล่นสามารถใส่เข้ามาในตัวเกม
- 19 Random Explosion:** การเพิ่มระเบิดเข้าไปในเกม ในเคสที่ยกตัวอย่างมาอย่าง 2D Side scroller นั้น ช่วยเพิ่ม action บนหน้าจออย่างมาก ทำให้ผู้เล่นต้องจดจ่ออยู่กับสิ่งที่เกิดขึ้น
- 20 Faster Enemies:** เมื่อได้ดีไซน์ให้ผู้เล่นมีความสามารถในการกำจัดศัตรูได้รวดเร็ว ก็ควรที่จะเพิ่มความเร็วของศัตรูด้วยเพื่อให้สมดุลกัน
- 21 Meaning:** เกมที่ดีไซน์ควรมีแรงจูงใจแก่ผู้เล่น ไม่ว่าแรงจูงใจนั้นจะเป็นอะไรก็ตาม

## 2.3 โปรแกรมสร้างเอฟเฟ็คเสียงอย่างง่าย Sfxr และ Bfxr

Sfxr เป็นโปรแกรมสำหรับสร้างเสียงประกอบในเกมอย่างง่าย โดยตัวโปรแกรมนั้นมีเสียงให้เลือกใช้หลายแบบ เราสามารถสร้างเสียงตามรูปแบบที่เราต้องการ มีปุ่มให้เลือกชนิดของเสียง ปรับความถี่ ประเภทคลื่นเสียงต่างๆ เก็บเสียงในรูปแบบไฟล์ .wav



รูปที่ 2.3 โปรแกรม Sfxr

Bfxr เป็นเวอร์ชันพัฒนามาจาก Sfxr มีฟังก์ชันการใช้งานให้เลือกได้มากขึ้น ให้เสียงที่มากขึ้น สามารถสร้างเสียงหลายเสียงพร้อมกันเพื่อฟังเปรียบเทียบได้ และมีระบบ Mixer เพิ่มเข้ามาเพื่อให้ผู้ใช้ได้ผสมเสียงใหม่ขึ้นเองได้อีกด้วย



รูปที่ 2.3 โปรแกรม Bfxr

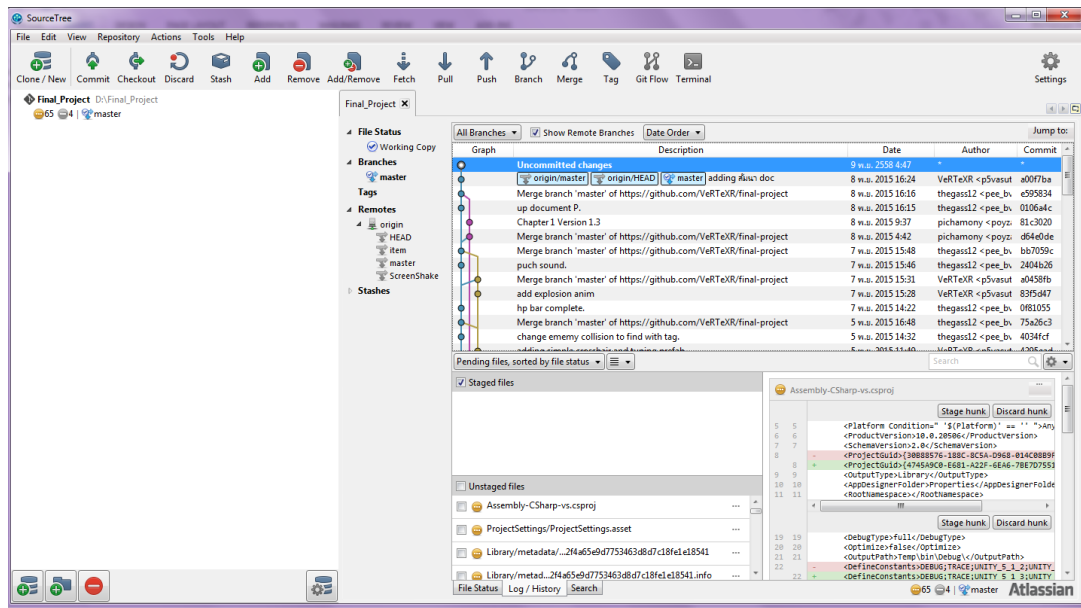
## 2.4 ควบคุม Version Control ผ่านการใช้งานโปรแกรม Source Tree ร่วมกับ GitHub

ผู้พัฒนาได้เลือกใช้ระบบ Git ซึ่งเป็น Version Control รูปแบบหนึ่ง มีข้อดีในการทำงานร่วมกันเป็นทีม สามารถควบคุมและอัปเดตปรับปรุงงานได้ โดยที่ทราบว่ามีแต่ครั้งคนในทีมเปลี่ยนแปลง แก้ไขอะไรไปบ้าง ถ้าหากว่ามีข้อผิดพลาด เวอร์ชันแรกนั้นดีกว่าล่าสุด ก็สามารถกด Revision กลับมาได้

Source Tree เป็นซอฟต์แวร์ที่ช่วยจัดการกับ Git repository ได้ง่ายขึ้น มี GUI ที่ใช้งานง่าย สะดวกกว่าควบคุมผ่านระบบ command line



รูปที่ 2.4 โลโก้โปรแกรม SourceTree

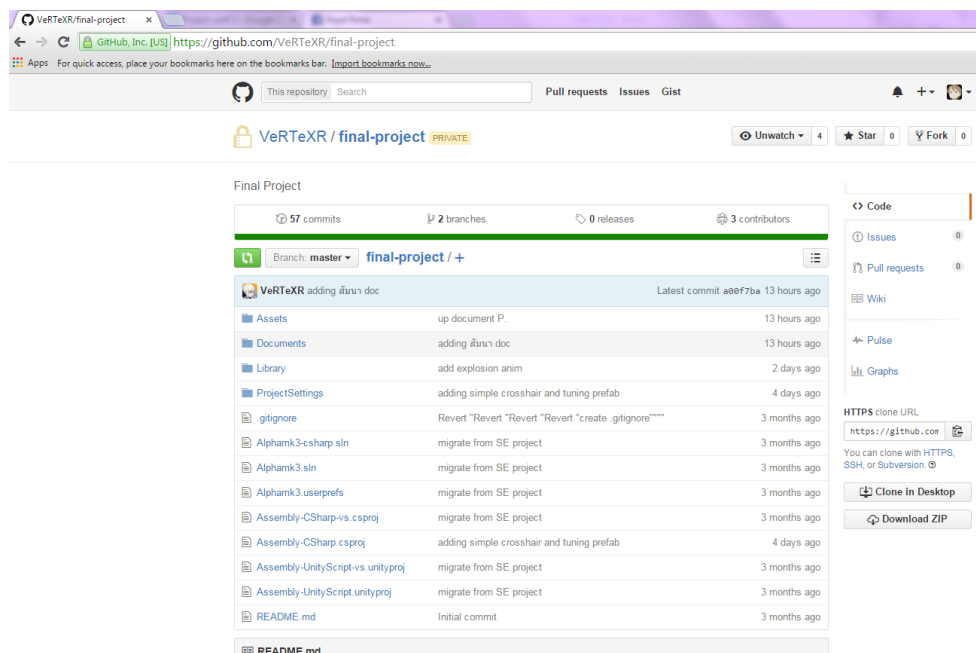


รูปที่ 2.5 โปรแกรม SourceTree

GitHub เป็นเว็บเซิร์ฟเวอร์ที่ให้บริการในการฝากไฟล์ Git โดยเราใช้เพื่อเก็บโปรเจกต์งานร่วมกันไว้ที่นี้ และใช้งาน อัปเดต แก้ไขผ่าน SourceTree

# GitHub

รูปที่ 2.6 โลโก้ Github



รูปที่ 2.7 ไฟล์ต่างๆ ที่อยู่บน GitHub

## 2.5 การควบคุมความเร็วของศัตรูของเกมแนว Shoot em' Up (Enemy Speed Control on Shoot em' Up Game With Fuzzy Takagi Sugeno Method)

Shoot em' Up Game เป็นประเภทย่อยประเภทหนึ่งของเกม ต่อสู้ ซึ่งเป็นเกมที่เป็นที่นิยมด้วยการมี ส่วนปฏิสัมพันธ์ที่น่าสนใจ ด้วยกับการที่เกมนี้มีจุดมุ่งหมายในการเล่นคือ การกำจัดศัตรูให้หมดไป ซึ่งเกมชนิดนี้อาจทำให้ผู้เล่นรู้สึกเบื่อหน่ายหากว่าศัตรูนั้นมีพฤติกรรมเคลื่อนไหวที่ไปในทิศทางเดียว ดังนั้น เกมชนิดนี้จึงต้องการตัวควบคุมสำหรับควบคุมการเคลื่อนไหวที่มีการเปลี่ยนแปลงอยู่เสมอของศัตรู เช่น ปัญญาประดิษฐ์ ดังนั้น งานวิจัยนี้จึงได้นำเสนอ กระบวนการฟัซซี่ของ ทาคากิ ซูกินะ ซึ่งจะเป็นตัวจัดการพฤติกรรมเคลื่อนไหวของศัตรูในเกมเพื่อทำให้เกมมีความน่าสนใจมากยิ่งขึ้น

เกมที่ดี ไม่ใช่เกมที่ให้ความสุขต่อผู้เล่นเพียงอย่างเดียว แต่ต้องเป็นเกมที่ให้ประโยชน์ต่อผู้เล่นด้วย และหนึ่งในประโยชน์นั้นก็คือ การที่ผู้เล่นได้มีการพัฒนาสมอง ซึ่งเกมเป็นระบบหนึ่งที่ผู้เล่นสามารถเป็นส่วนหนึ่งในการควบคุมและเผยแพร่วัฒนธรรมได้ ซึ่งผู้เล่นจะมีปฏิสัมพันธ์กับระบบ และ ความขัดแย้งต่างๆในเกม ในรูปแบบของวิศวกรรมเสมือน

Shoot em' Up Game เป็นเกมแนวยิงกันที่สามารถเล่นได้แบบผู้เล่นสองคนเล่นกัน หรือ แบบผู้เล่นคนเดียวเล่นกับศัตรูที่เป็นปัญญาประดิษฐ์ซึ่งการเล่นเกมนี้นี้เป็นไปแบบง่ายๆคือ ผู้เล่นจะต้องกำจัดศัตรูให้หมดและในขณะเดียวกันก็ต้องพยายามเอาตัวรอดจากการโจมตีของศัตรูด้วยเช่นกัน ซึ่งในงานวิจัยนี้จะนำเสนอเรื่องของความเร็วในการเคลื่อนที่ของยานอวกาศในขณะที่ยานลอยอยู่และทำการโจมตีผู้เล่น เพื่อสร้าง

ให้เกมดูน่าสนใจมากยิ่งขึ้น ศัตรูในเกมจึงถูกสร้างให้เป็นปัญญาประดิษฐ์ และโดยเฉพาะเวลาในการตอบสนองของศัตรูนั้นสามารถคาดการณ์ได้ สามารถหลบหลีก และจดจำพฤติกรรมเคลื่อนไหวของศัตรูได้ Shoot em' Up Game เป็นเกมที่จะมีเป้าหมายในการออกแบบศัตรูให้ช่วยปัญญาประดิษฐ์ในการควบคุมระดับความยากของเกม

Fuzzy Takagi – Sugeno เป็นตัวช่วยในการควบคุมความเร็วของศัตรูซึ่งจะใช้ตัวแปรต่างๆในเกมที่รับเข้ามาเพื่อประมวลผลและส่งออกค่าสิ่งที่จะขึ้นอยู่กับตัวแปรที่นำเข้ามาด้วย ซึ่งจะทำให้พฤติกรรมเคลื่อนไหวของศัตรูนั้นไม่เป็นไปในทิศทางเดียวหรือเส้นตรง

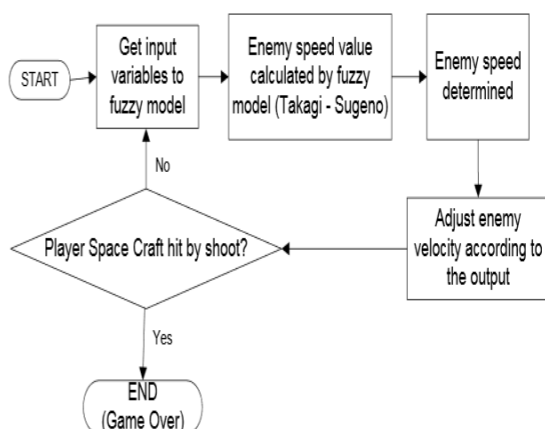
Fuzzification เป็นขั้นตอนการเปลี่ยน ค่าที่ได้รับจากตัวแปรต่างๆเข้าไปเก็บในเซตของเงื่อนไขต่างๆ เพื่อนำไปใช้ในการทำงานต่อไป ซึ่งขั้นตอนในการทำ Fuzzification มีดังนี้

1. รับค่าจากตัวแปรที่นำเข้ามา
2. ดำเนินการจัดเก็บรวบรวมข้อมูลที่ได้รับมาจากตัวแปรต่างๆที่สอดคล้องกัน
3. ดำเนินการแปลงค่าที่ได้ให้ชัดเจนเพื่อส่งค่าออกไปทำงาน

Defuzzification เป็นการหาค่าฟัซซี่ให้เป็นค่าที่ใช้งานจริง ซึ่งในการนำเสนองานวิจัยนี้ใช้สมการในการแปลงค่าดังต่อไปนี้  

$$x^* = \frac{\sum_{i=1}^n m^i w_i}{\sum_{i=1}^n m^i}$$
 $m^i$  เป็นค่าเอาท์พุทที่ได้ของกฎแต่ละข้อ  
 $w_i$  เป็นค่าเฉลี่ยถ่วงน้ำหนักของกฎแต่ละข้อ

การนำเสนองานวิจัยในครั้งนี้ ฟัซซี่ ลอจิก ที่ใช้ในการคำนวณความเร็วในการเคลื่อนที่ของศัตรูจะอธิบายโดย flow chart นี้



รูปที่ 2.8 Flow Chart แสดงการทำงานของ

ของ กระบวนการฟัซซี่ของ ทาคากิ ซูเกโนะ

จากภาพจะแสดงการทำงานของ Fuzzy Takagi – Sugeno โดยเริ่มจากการรับค่าต่างๆจากตัวแปร โดยค่าที่รับจะมี ดังนี้

- Distance : ระยะห่างระหว่างตัวศัตรู กับ ผู้เล่น
- Enemy Health Point : พลังชีวิตของศัตรู
- Player Remaining Life : พลังชีวิตที่เหลืออยู่ของผู้เล่น



โดยเมื่อได้ค่าทั้งหมดแล้ว จะนำไปเปรียบเทียบกับค่าในตารางเปรียบเทียบเพื่อหาค่าที่จะส่งกลับไปควบคุมความเร็วในการเคลื่อนที่ของศัตรู ซึ่งตารางเปรียบเทียบเพื่อคำนวณค่าความเร็วมีดังนี้

ตารางที่ 1 ตารางแสดงรายการกฎของฟัซซี่

TABLE 1. LIST OF THE FUZZY IF-THEN RULES

R1:	IF distance is close and enemy HP is low THEN enemy speed is fast
R2:	IF distance is close and enemy HP is high THEN enemy speed is medium
R3:	IF distance is medium or enemy HP is low THEN enemy speed is slow
R4:	IF distance is medium or enemy HP is high THEN enemy speed is medium
R5:	IF distance is far and enemy HP is low THEN enemy speed is slow
R6:	IF distance is near or player remaining life is low THEN enemy speed is fast
R7:	IF distance is near or player remaining life is medium THEN enemy speed is fast
R8:	IF distance is medium and player remaining life is low THEN enemy speed is fast
R9:	IF distance is medium or player remaining life is medium THEN enemy speed is medium
R10:	IF distance is medium or player remaining life is high THEN enemy speed is slow
R11:	IF distance is far or player remaining life is high THEN enemy speed is slow
R12:	IF enemy HP is low and player remaining life is low THEN enemy speed is fast
R13:	IF enemy HP is low or player remaining life is high THEN enemy speed is slow
R14:	IF enemy HP is medium and player remaining life is high THEN enemy speed is slow
R15:	IF enemy HP is high or player remaining life is low THEN enemy speed is fast
R16:	IF enemy HP is high or player remaining life is medium THEN enemy speed is medium

ดังในตาราง ใช้การดำเนินการโดยใช้ AND และ OR ในการดำเนินการเปรียบเทียบค่าต่างๆ โดย AND จะใช้ดำเนินการกับค่าขนาดเล็ก และ OR จะใช้ดำเนินการกับค่าขนาดใหญ่

จากการศึกษางานวิจัยในครั้งนี้ทำให้ผู้ศึกษาได้เรียนรู้วิธีการจัดการพฤติกรรมเคลื่อนที่ของศัตรูในเกมแนว Shoot em' Up Game ซึ่งการจัดการความเร็วในการเคลื่อนที่ของศัตรู ทำให้เกมมีความน่าสนใจ และ ทำทายมากยิ่งขึ้น ทั้งยังทำให้ผู้เล่นสามารถฝึกการใช้สมองได้อีกด้วย ไม่เพียงแค่การจดจำรูปแบบการเคลื่อนไหวเดิมๆของศัตรูอีกต่อไป

## 2.6 ทฤษฎีฟลอว์ วิวัฒนาการ และ ความคิดสร้างสรรค์: หรือ ‘ความสนุก และ เกม’

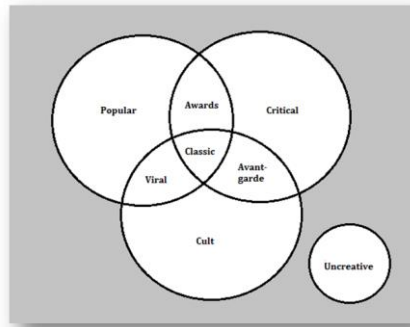
เกมและ สื่อต่างๆที่เกี่ยวข้อง ในมุมมอง การสร้าง และ การตอบรับของผู้เล่นโดยใช้ระบบโมเดลความคิดสร้างสรรค์ DPFi เพื่อที่จะแสดงให้เห็นถึงวิวัฒนาการของสื่อที่เกิดขึ้น ความสัมพันธ์ที่ส่งผลต่อกันและกัน ทฤษฎีฟลอว์ของความคิดสร้างสรรค์เป็นสิ่งชี้ให้เห็นถึง ‘fun factor’ ของเกม และ ทฤษฎี ‘Narrative Transportation’ ชีวัดในส่วนหนึ่งของเรื่องราว รวมถึงความสัมพันธ์ของทฤษฎีข้างต้น กับ ‘เนื้อเรื่อง’ ของเกม นอกจากนั้นยังใช้ทฤษฎีความคิดสร้างสรรค์ทั่วไปของ Boyd (2009) ว่า ศิลปะ คือ ‘การเล่นกับแพทเทิร์น’ ดังนั้นจึงสามารถยกบรรทัดที่ว่า เกมเพลย์ ไม่ว่าจะรูปแบบใดก็ตามสามารถที่จะส่งเสริมพัฒนาการความฉลาดของสัตว์ได้ ดังนั้นเกมในฐานะของศิลปะอย่างหนึ่งนั้นอาจจะสามารถที่จะเพิ่มความฉลาดให้กับมนุษย์ได้

ทฤษฎีฟลอว์ ในปี 1996 Csikszentmihalyi ได้นำเสนอปัจจัยต่างๆที่เป็นเอกลักษณ์ของ ฟลอว์ ขึ้นมาได้แก่: (1) ในการกระทำใดการกระทำหนึ่งมีเป้าหมายอย่างชัดเจนเป็นขั้นตอน (2) เกิดผลของการกระทำทันทีที่มีการกระทำเกิดขึ้น (3) ความยากของเป้าหมายและความสามารถของผู้ทำงานนั้นๆเหมาะสมกัน (4) การกระทำและการรับรู้ของผู้กระทำนั้นได้รวมเป็นหนึ่งเดียวกัน (5) สิ่งรบกวนทั้งหลายถูกตัดออกจากความคิด (6) ไม่มีความกลัวที่จะล้มเหลว (7) ผู้กระทำนั้นไม่คำนึงถึงสิ่งที่ตัวเองเป็น (8) การรับรู้เวลาของผู้กระทำนั้นถูก

บิตเพื่อนไป (9) การกระทำนั้นๆกลายเป็นงานอัตโนมัติ โดยในงานวิจัยของ Sweetser & Wyeth นั้นได้แสดงให้เห็นว่า เกมที่ได้รับการยอมรับว่าเป็นเกมที่ ‘สนุก’ นั้นมีคุณสมบัติเหล่านี้ทั้งหมด การติดอยู่ใน โพลว์ นั้นเท่ากับว่ากำลัง สนุก กับการเล่นเกม – หรือ สามารถใช้สุภาษิตที่ว่า “เวลามักผ่านไปเร็วเสมอเมื่อตนเองรู้สึกสนุก” ได้อย่างชัดเจน โดย Csikszentmihalyi ได้นำเสนอโมเดลของ ‘โพลว์’ โดยสามารถเห็นได้ชัดว่าจะสามารถเกิดขึ้นได้เมื่อ ความสามารถของผู้เล่นเหมาะสมกับการท้าทายที่ได้เผชิญ จะทำให้ผู้เล่นนั้นๆตกอยู่ใน ‘โพลว์’

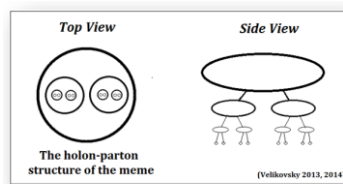
**ทฤษฎี Narrative Transportation** ในเกมส่วนใหญ่แล้วจะมีเรื่องราวที่คอยส่งเสริมเกมเพลย์ของเกมนั้นๆ อยู่ เพื่อที่จะช่วยให้ผู้เล่นสามารถ เป็นส่วนหนึ่งเข้ากับเกมได้ง่ายขึ้น เรื่องราวที่วนนี้คือสิ่งที่ทำให้เกิด Narrative Transportation โดย Van Laer ได้ให้นิยามไว้ว่า ‘Narrative Transportation คือกระบวนการที่ ผู้บริโภคสื่อ ถูกดูดเข้าไปในเนื้อเรื่องที่บริโภค โดยจะทำให้เนื้อเรื่องที่เล่านั้นเสมือนเกิดขึ้นจริงใน ความคิด และ จิตใจ ของผู้บริโภคสื่อ นั้นเมื่อ เนื้อเรื่องนั้นๆ หรือ ประสบการณ์ส่วนบุคคลของผู้บริโภคเข้ากันได้ ส่วนในปี 2002 Green and Brock ได้ให้นิยามไว้ว่า Narrative Transportation นั้นเกิดขึ้นเมื่อ ผู้บริโภคสื่อ รู้สึกเหมือนได้เข้าไปอยู่ในโลกที่ถูกสร้างขึ้นในเรื่องราวที่เกิดขึ้น เพราะมีความผูกพันกับตัวละคร และ จินตนาการ ที่วาดพล็อตเรื่องนั้นๆไว้

**ความเกี่ยวข้องระหว่างนิยาม ‘ความคิดสร้างสรรค์’ และ โมเดล DPFi** ระบบ DPFi (Domain, Person, Field Interactions) นั้นเป็นระบบที่เชื่อมระหว่างวิทยาศาสตร์และศิลปะ โดยนิยามที่เรียบง่ายที่สุดของ ‘ความคิดสร้างสรรค์’ นั้นมาจากงานวิจัยด้านจิตวิทยาซึ่งสามารถนิยามออกมาได้ว่า ‘ความคิดสร้างสรรค์’ คือ ไอเดีย กระบวนการ หรือ ชิ้นงาน ที่ ‘ใหม่ และ เหมาะสม’ โดยในปี 2006 Csikszentmihalyi ได้อธิบายระบบโมเดลของความคิดสร้างสรรค์ไว้ดัง รูปที่ 1 “เพื่อที่จะให้เกิดความสร้างสรรค์ขึ้นมา บุคคลต้องสร้างผลิตภัณฑ์ที่เป็นสิ่งแปลกใหม่ไปจากที่มีอยู่ในโดเมน จากนั้นผลิตภัณฑ์จะถูกเลือกโดยฟิลด์เพื่อที่จะให้เข้าไปอยู่ในโดเมน การสร้างสรรค์นั้นเกิดขึ้นเมื่อคนได้สร้างการเปลี่ยนแปลงในโดเมนที่จะถูกส่งต่อไปในเวลาต่อไป ” โดยกระบวนการข้างต้นนั้นเรียกได้ว่าเป็น อัลกอริทึมแห่งพัฒนาการ - มีการคัดเลือก การเปลี่ยนแปลง การส่งต่อข้อดี และ ข้อเสีย ดังนั้นเมื่อนำมาใช้กับเกม ถ้าหากว่าคนส่วนใหญ่ในฟิลด์เกมนั้นได้เห็นพ้องกัน ว่าเกมใหม่ที่ออกมานั้น สนุก และ สร้างสรรค์ (ใหม่ และ เหมาะสม) เกมนั้นก็จะเป็นเกมที่ได้รับการยอมรับ

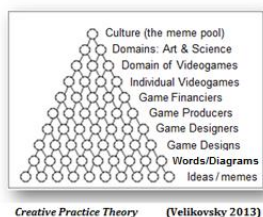


รูปที่ 2.9 ประเภทของความคิดสร้างสรรค์ หรือ ‘การยอมรับ’ ของวงการเกม: © JT Velikovsky

Holon-Parton เป็นหน่วยวัดค่าทางวัฒนธรรม - Arthur K ได้ให้คำนิยาม ‘holon’ ไว้ว่า สิ่งใดที่เป็นส่วนหนึ่งแต่ก็ เติบโตด้วยตัวเองในเวลาเดียวกัน นักฟิสิกส์ควอนตัม Richard P. ได้ใช้คำว่า ‘parton’ ในการเรียกคอนเซ็ปต์เดียวกันในทางฟิสิกส์ เช่นเดียวกันกับการแข่งขันกันระหว่างสิ่งมีชีวิตต่างสายพันธุ์ ในด้านวัฒนธรรมเกม (ในฐานะ มีม) ก็ต้องแข่งขันใน ‘สภาพแวดล้อม’ เดียวกันกับเกมอื่นๆ เพื่อเงินและความสนใจของผู้เล่น ในขณะเดียวกันก็อยู่ภายใต้ประเภทเดียวกัน และยังพยายามที่จะควบคุมและให้คำสั่งแก่ความสนใจของผู้เล่นเกม ในฐานะ holon-partons เป็นกลุ่มก้อนของไอเดีย ขั้นตอน และ ผลลัพธ์ (หรือ มีม) สามารถเห็นได้ชัดว่า มีมนั้นอยู่ทั้งชั้นบนสุดและล่างสุด แทนการทำซ้ำในตัวเองเมื่อเวลาผ่านไปเรื่อย อัลกอริทึมของการพัฒนาการ การคัดเลือก, แปรแยก (ประกอบด้วย การรวมกัน, กลายพันธุ์, และ ดัดแปลง) และ การส่งต่อโดยพันธุกรรม นั้นเกิดขึ้นใน ลำดับชั้นด้านล่างในฐานะระบบต่อยอดและย้อนทำซ้ำในตัวเองอย่างไม่ลืเมีย



รูปที่ 2.10 รูปแบบ Holon-Parton ของมีม, หน่วยทางวัฒนธรรม © JT Velikovsky



รูปที่ 2.11 ลำดับชั้นของโดเมนเกมและ ลำดับชั้นของมีม © JT Velikovsky

‘Transmedia Storytelling (การเล่าเรื่องข้ามสื่อ)’ นั้นถูกนิยามโดย Kinder, Jenkin และ ในกฎหมายโดย ว่า ‘เนื้อเรื่องที่เกี่ยวข้อกัน 3 เรื่องเป็นอย่างต่ำโดยเนื้อเรื่องทั้งหมดนั้นอยู่ในจักรวาลเดียวกันบนแพลตฟอร์มต่างกันไป’ โดยสามารถสังเกตได้ว่า ‘สื่อที่ถูกเล่าใหม่’ นั้นไม่ใช่เพียงแค่การแปลง เนื้อเรื่องเดิมไปบนแพลตฟอร์มอื่น แต่เป็นเนื้อเรื่องแยกไปที่มีบางจุดที่คล้ายคลึง และ บางจุดแตกต่างเป็นเอกลักษณ์อยู่บนแพลตฟอร์มของตัวเอง ดังนั้น กฎของอัลกอริทึมในการพัฒนาการ (และ ลำดับชั้น) จะสามารถนำมาใช้ได้เช่นเดียวกันกับที่มีการใช้ในเกม

กระบวนการของการดีไซน์เกมเองก็สามารถมองได้ว่าเป็นการ ‘เล่นกับแพทเทิร์น’ เมื่อ เกมดีไซน์เนอร์สร้างแพทเทิร์นของเกมเพลย์ ถ้าแพทเทิร์นเหล่านั้นนั้นไม่มีความแตกต่าง จากเกมที่ประสบความสำเร็จก่อนหน้านี้มากพอ เกมนั้นก็就会被ตราหน้าว่า ‘ไม่มีความเป็นตัวเอง’ (หรือ ไม่มีความคิดสร้างสรรค์) - ในการเล่นซ้ำ ผู้เล่นเองก็สามารถที่จะเปลี่ยนแพทเทิร์นของการเดินทางของตนเองในเลเวลนั้นๆของเกมได้เพื่อที่จะปรับกลยุทธ์ตัวเองเพื่อที่จะสามารถเล่นเกมจบได้เร็วที่สุด

ในงานวิจัยของ Capra & Luisi (2014) นั้นได้นิยามว่า ‘ความฉลาด’ คือ ‘ความสามารถในการแก้ไขปัญหา’ เกมส่นั้นได้เพิ่มความฉลาดแก่ผู้เล่นได้มาก เนื่องจากผู้เล่นนั้นต้องแก้ปัญหาใหญ่นั้นคือ เป้าหมายหลักของด่าน และ เป้าหมายย่อยทั้งหมด โดยจากงานวิจัยของมหาวิทยาลัย Queen Mary University of London และ University College London (UCL) ได้ทดสอบการตอบสนองต่อสิ่งเร้าของอาสาสมัครจำนวน 72 คน โดยได้มีการให้อาสาสมัครนั้นเล่นเกม Starcraft และ The Sims โดยผลสรุปออกมานั้น พบว่า เกมส่บางประเภทนั้นสามารถเพิ่มความฉลาดให้กับผู้เล่นได้มากกว่าเกมส่ประเภทอื่นๆ โดย Starcraft นั้นสามารถเพิ่มความฉลาดให้กับผู้เล่นได้มากกว่า The Sims นอกจากนั้น ในปี 2014 งานวิจัยของ Przybylski et al ได้พบว่าเกมส่ที่ถูกดีไซน์มาแย่ สามารถที่จะเพิ่มความก้าวร้าวให้กับผู้เล่นได้ แต่อย่างไรก็ตามเกมส่นั้นๆก็เป็นตัวเปิดโอกาสให้กับผู้เล่นได้ เรียนรู้วิธีการที่จะเผชิญหน้ากับดีไซน์ที่ย่ำแย่เหล่านั้น ทำให้เกิดการพัฒนาด้านอารมณ์ในตัวผู้เล่น ดังนั้นเราจึงสามารถกล่าวได้ว่า เกมส่ที่แย่นั้นก็สามารถที่จะทำให้ ความฉลาดด้านอารมณ์ของผู้เล่นนั้น พัฒนาขึ้นเช่นกัน

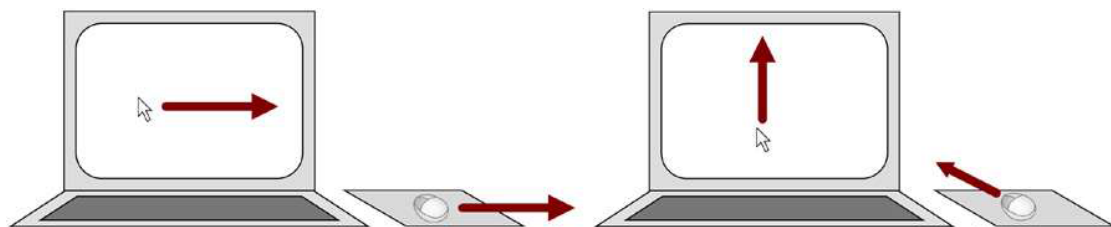
ในการพัฒนาเกม ดีไซน์เนอร์ มักเลือกไอเดียที่ดีที่สุดจาก ที่มีอยู่ในปัจจุบัน และ นำมารวมกับไอเดียอื่น หรือ ไอเดียของตนเองมีอยู่ เพื่อให้เกิดเป็น ไอเดียใหม่ที่แตกต่างจากเดิมขึ้น โดยในมุมมองนี้ ความคิดสร้างสรรค์ทั้งหมด ได้เกิดมาจากอัลกอริทึมการพัฒนาเช่นนี้ โดยมีแรงกดดันจากการเลือกปะทะกับไอเดียใน

ทุกชั้น และ ทุกลำดับชั้นของ holon-parton ถ้าหากว่าไม่ใช่เช่นนั้น เกมนั้นๆสามารถพูดได้ว่าเป็นเกมที่ไม่เต็มเกม และ สามารถพิจารณาได้ว่า ถูกดีไซน์มาไม่ดีพอ

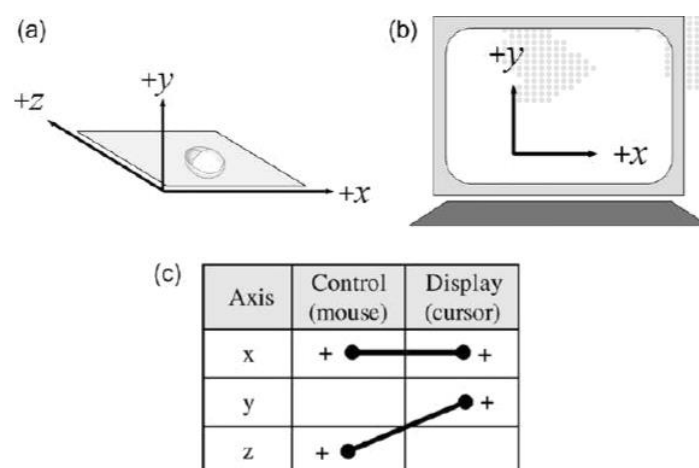
## 2.7 ความสัมพันธ์ระหว่างการควบคุมและมุมมองที่เห็น

ใช้หลักการของปฏิสัมพันธ์ของมนุษย์กับคอมพิวเตอร์มาช่วยในการออกแบบ Interface, การเคลื่อนไหว, ควบคุมเกม เพื่อให้ผู้เล่นได้รู้สึกว่าการเล่นง่ายและสิ้นเปลืองมากที่สุด

ความสัมพันธ์ในพื้นที่ว่างนั้น ขึ้นอยู่ระหว่างการควบคุมของผู้เล่นจากอุปกรณ์เล่นเกม (Hard Control) และภาพที่มองเห็นในหน้าจอ (mapping) พื้นที่ของเมาส์ที่ใช้ควบคุมนั้นประกอบด้วยแกน x, y, z ส่วนพื้นที่ในหน้าจอประกอบด้วยแกน x และ y เท่านั้น ดังแสดงในรูปที่ 2.13



รูปที่ 2.12 แสดงให้เห็นทิศทางการเคลื่อนที่ของเมาส์ที่คนบังคับ กับเมาส์ที่เคลื่อนไหวบนหน้าจอ



รูปที่ 2.13 (a) พื้นที่ควบคุม (b) พื้นที่แสดงผล (c) แสดงถึงความสัมพันธ์ของการควบคุมเมาส์และลูกศรบนหน้าจอตามแกน x, y, z